

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

THIS PAGE BLANK (USPTO)



PATENT ABSTRACTS OF JAPAN

(11) Publication number: 2001222429 A

(43) Date of publication of application: 17.08.01

(51) Int. Cl

G06F 9/44

(21) Application number: 2000031688

(71) Applicant: FUJITSU LTD

(22) Date of filing: 09.02.00

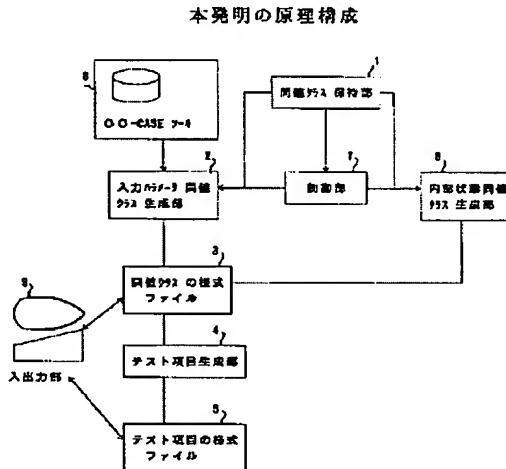
(72) Inventor: OHASHI KYOKO
YAMAMOTO RIEKO

(54) TEST ITEM GENERATION SUPPORTING DEVICE COPYRIGHT: (C)2001,JPO

(57) Abstract:

PROBLEM TO BE SOLVED: To support simple work and a work with many reference items and to generate only practically sufficient test items for a set type being one of the factors of the increase of test items in a test items generation supporting device for supporting a test work based on object directional design.

SOLUTION: This device is provided with an equivalence class holding part for inputting and holding the equivalence class for every object, an input parameter equivalence class generating part for reading an object directional CASE tool based on UML semantics and the information of the equivalence class from the equivalence class holding part, and for reading the signature of one method, and for generating the equivalence class of the input parameter of a method to be tested, and a test item generating part for generating a test item from the equivalence class. Then, all the equivalence classes of the input parameters of methods to be tested including inheritance relations are described, and test items are generated based thereon.



THIS PAGE BLANK (USPTO)

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2001-222429

(P2001-222429A)

(43)公開日 平成13年8月17日 (2001.8.17)

(51)Int.Cl.⁷

G 0 6 F 9/44

識別記号

5 3 0

F I

G 0 6 F 9/44

テ-マコ-ト^{*}(参考)

5 3 0 P

審査請求 未請求 請求項の数4 OL (全26頁)

(21)出願番号

特願2000-31688(P2000-31688)

(22)出願日

平成12年2月9日 (2000.2.9)

(71)出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番
1号

(72)発明者 大橋 恭子

神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(72)発明者 山本 里枝子

神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(74)代理人 100094662

弁理士 穂坂 和雄 (外2名)

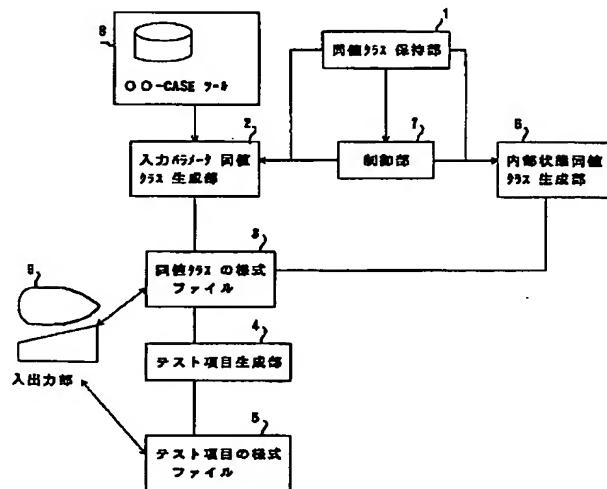
(54)【発明の名称】 テスト項目生成支援装置

(57)【要約】

【課題】本発明はオブジェクト指向設計に基づくテスト作業の支援を行うためのテスト項目生成支援装置に関し、単純な作業、参照項目の多い作業を支援し、テスト項目が増大する原因の一つである集合型に対して実用上十分なテスト項目のみを生成することを目的とする。

【解決手段】オブジェクト毎にその同値クラスを入力保持する同値クラス保持部と、UMLセマンティックス準拠のオブジェクト指向CASEツールと同値クラス保持部から同値クラスの情報を読み出して、一つのメソッドのシグネチャを読み出して、テスト対象となっているメソッドの入力パラメータの同値クラスを生成する入力パラメータ同値クラス生成部と、同値クラスからテスト項目を生成するテスト項目生成部とを備え、継承関係を含むテスト対象のメソッドの入力パラメータの全同値クラスを記述して、それに基づいてテスト項目を生成するよう構成する。

本発明の原理構成



【特許請求の範囲】

【請求項1】 オブジェクト指向設計に基づくテスト作業の支援を行うためのテスト項目生成支援装置において、オブジェクト毎にその同値クラスを入力保持する同値クラス保持部と、UMLセマンティックス準拠のオブジェクト指向CASEツールと前記同値クラス保持部から同値クラスの情報を読み出して、一つのメソッドのシグネチャを読み出して、テスト対象となっているメソッドの入力パラメータの同値クラスを生成する入力パラメータ同値クラス生成部と、同値クラスからテスト項目を生成するテスト項目生成部と、前記入力パラメータ同値クラス生成部は、継承関係を含むテスト対象のメソッドの入力パラメータの全同値クラスを決められた様式で出力可能に記述し、前記テスト項目生成部は前記様式で記述された入力パラメータ同値クラスに基づいてテスト項目を生成することを特徴とするテスト項目生成支援装置。

【請求項2】 請求項1において、前記入力パラメータ同値クラス生成部は、メソッドの入力パラメータがオブジェクトである場合に、他のオブジェクトと集約関係にある場合にそのオブジェクトも入力パラメータに加え、前記テスト項目生成部においてテスト項目を生成することを特徴とするテスト項目生成支援装置。

【請求項3】 請求項2において、前記入力パラメータ同値クラス生成部は、集約関係をたどる際に関連するメソッドの向きを表すことを定義されたナビゲーションを参照し、参照できる向きの関連先のオブジェクトのみを入力パラメータに加えることを特徴とするテスト項目生成支援装置。

【請求項4】 請求項1において、UMLセマンティックス準拠のオブジェクト指向CASEツールからテスト対象となっているメソッドが含まれるシーケンス図を検索し、検索されたシーケンス図に現れるオブジェクトから入力パラメータと関連オブジェクト以外のオブジェクトを検出して、メソッドが動作する時に必要なオブジェクトの候補とし、当該オブジェクトの同値クラスを生成する内部状態同値クラス生成部を設け、前記内部状態同値クラス生成部から出力される同値クラスは、前記入力パラメータの同値クラスと同様の様式で出力され、前記テスト項目生成部からは前記様式の同値クラスからテスト項目を出力し、入力パラメータのテスト項目と内部状態のテスト項目を組み合わせることを特徴とするテスト項目生成支援装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明はオブジェクト指向設計に基づくテスト作業の支援を行うためのオブジェクト指向のテスト項目生成支援装置に関する。

【0002】 ソフトウェアが様々な分野で使われ、且つその規模が増大するに従って、そのバグによる影響が重

大になりつつある。そこで、テスト作成の工数をできるだけ減らすことと、バグを予め取り除くことと、必要なテストが必ず行われることが要求されている。

【0003】

【従来の技術】 従来、アプリケーションソフトの機能テストは機能仕様に基づいてテスト担当者が設計していた。テストの設計は開発後期になってから行われるため、どうしてもテスト作業全般に工数が割けられないという事態になる。また、テストの設計には、対象アプリケーションの理解や、経験が必要なため、経験豊富な開発者がテスト設計に従事しなければならなかった。

【0004】 このため、経験不足の者がテストの設計をすると、アプリケーションの動作が変化する条件の考え方落としや、条件の組み合わせのバリエーションが十分に作られない事態が生じる。

【0005】 近年、オブジェクト指向言語が出現した後、オブジェクト指向プログラミングのためにオブジェクト指向設計が必要となり、そして、オブジェクト指向設計をするためにオブジェクト指向分析が必要となった。このようなオブジェクト指向を各業種や業務に適用するための方法論が種々提案されているが標準化されてなく、モデルの表記法についてUML (Unified Modeling Language) としてOMG (Object Management Group) において標準化された。

【0006】 オブジェクト指向における、オブジェクトはデータ（状態、情報、データ構造）とデータを処理するための幾つかの操作を保持している。また、同じ性質を持つオブジェクトを集めて抽象化したものをクラスといい、クラスは名称、属性及び操作とで構成される。オブジェクトの抽象化にはデータ抽象化とスーパー抽象化の2つのレベルがありデータ抽象化はオブジェクトからクラスを作成するときの抽象化で、主にオブジェクトが個々に保有している具体的な値が偶然的な性質として省かれ、システムにとって根本的でない性質（属性）も省かれれる。それに対し、スーパー抽象化は、複数のクラスの中で同じ性質（対象システムにとって根本的な性質が同じ）を持ったクラス同士を更に抽象化することをいう。スーパー抽象化したクラスをスーパークラスといい、その元となったクラスをサブクラスという。スーパークラスには、複数のクラスの共通的な性質を定義し、サブクラスには共通化できなかった固有の性質だけを残して定義する。

【0007】 このようなオブジェクト指向によるアプリケーションソフトをテストする場合、テスト対象のメソッドだけでなく、関連したオブジェクトやスーパークラスを考慮しなければならないので、更に複雑になっていた。

【0008】 また、テスト技法の一つとして同値分割の手法がある。この手法は、例えば、あるアプリケーションが、引数としてint(整数)を表す「i」とString(文

字列) を表す「s」を使用するものとし, int の i が $i < 0, 0 \leq i \leq 100, 100 >$ i の各条件のそれぞれに応じてアプリケーションが異なる演算を行う場合があり、このように i が分割されることを 3 つの場合に同値分割するという。このように同値分割されるアプリケーションの動作をテストする場合、3 つの各場合のそれぞれの代表例を一つ選んでテストすれば、その中のどの値についても正しく動作することがチェックしたことになる。この例では、引数として i だけでなく s もあり、s についてもアプリケーションの演算が変化する場合、i の値としてとり得る値の種類（この例では 3 種類）と s として取りえる値の種類（この例では 2 種類とする）の組み合わせ（この例では $3 \times 2 = 6$ 組）の処理でテスト項目を作成する。しかし、実際のアプリケーションでは多数の引数を使用しているため、単純に組み合わせ処理をすると、テスト項目数が膨大な数に上り、実際にはそれらすべてに対しテストケースを作成、実行することは時間及び作業が膨大となり極めて困難であった。

【0009】

【発明が解決しようとする課題】テスト設計作業には、オブジェクトの設計情報だけからでも可能な作業がある。すなわち、従来の UML セマンティックス (Semantics) 準拠 (UML により各表記の意味、解釈が規定されたものに準拠) の CASE (Computer Aided Software Engineering) ツールを用いてどのような引数があるかを見ることで設計作業に役立てることができるが、その作業によって作成されるテスト項目はアプリケーションのテストにおいて最低限必要なものであるが、単純で且つ、参照項目が多いため見落としが生じやすく作業者の負担が増え、作業ミスも起こりやすいという問題があった。

【0010】機能テスト以外では、全てのソースコードを必ず一回は実行するという方法もあるが、時間と手間がかかるという問題があった。

【0011】本発明は開発の時に作成した設計ドキュメントを利用して、単純な作業、参照項目の多い作業を支援し、テスト項目が増大する原因の一つである集合型（配列やリスト等、複数の要素を含む型）に対して実用上十分なテスト項目のみを生成することで、テスト項目を実行可能な程度に抑えることを可能とするテスト項目生成支援装置を提供することを目的とする。

【0012】

【課題を解決するための手段】図 1 は本発明の原理構成である。図中、1 はテストに必要な同値クラスを保持する同値クラス保持部、2 はテストしようとするメソッドへの同値クラスの入力パラメータを生成する入力パラメータ同値クラス生成部、3 は入力パラメータ同値クラス生成部 2 や内部状態同値クラス生成部 6 によって生成された同値クラスのデータが格納され出力が行われる同値クラスの様式ファイル、4 は列挙された同値クラスを組

み合わせることによってテスト項目を生成するテスト項目生成部、5 はテスト項目生成部 4 で生成したテスト項目の様式が格納されたテスト項目の様式ファイルである。6 は同値クラス保持部 1 からの情報を基にテスト対象メソッド（プログラム）内で必要となるオブジェクトやマスター等と、その同値クラスを列挙する内部状態同値クラスの生成部、7 は制御部、8 は対象となるメソッドのクラス名、引数名等がとれることを保証した UML セマンティックス準拠の OO-CASE (Object Oriented-Computer Aided Software Environment : オブジェクト指向ソフトウェア支援環境) ツール、9 は利用者への表示出力を提示したり、利用者による表示内容への編集、指示の入力を実行する入出力部である。

【0013】図 2 は同値クラス保持部で保持するデータの構成例である。同値クラスとは、アプリケーションが同じ処理で動く値の範囲を示しており、オブジェクトや型ごとに、同値クラスを与える。同値クラスを与える時に、どの同値クラスの値がアプリケーションの正常な処理を行わせるものであれば「正」のタグ、エラーを引き起こすものであれば「異」のタグをつけ、動作によって変わるものにはタグを付けない。これらのタグは、後の作業でテスト項目を減らす工夫をする時に利用する。図 2 の 10-1, 10-2, … 10-n の符号はそれぞれクラス名 1, クラス名 2, …, クラス名 N を表し、クラス名 1 (10-1) の場合、同値クラス 1-1 は「正」である範囲、同値クラス 1-2 は「異」（エラー）である範囲に対応するクラス名であり、10-m のクラス名 s-tr (文字列を表し) の場合、「null (内容が無い)」であれば「異」（エラー）であり、長さが 0 であれば「正」、長さが最大長であれば「正」とされる。

【0014】同値クラスは同値クラス保持テーブル（図示省略）からクラス名をキーにして、テスト設計者により作成される。アプリケーション毎に異なる部分もあるが、基本的な型の同値クラスはどのアプリケーションにも共通して使用できる。これらの同値クラスを用いてテスト項目を作成する。従って、ここで十分な同値クラスを生成しておくことで後の作業を軽減できる。

【0015】図 1 の作用を説明すると、同値クラス保持部 1 にテーブル形式で、図 2 に示すような同値クラスのデータを格納した状態で、入力パラメータ同値クラス生成部 2 の動作により UML セマンティックス準拠の CASE ツール 8 のクラス図から得られる情報を基にテスト対象メソッドの操作に対する入力パラメータの同値クラスを列挙した同値クラスのデータを取り出して同値クラスの様式ファイル 3 に格納する。この同値クラスの様式ファイル 3 の内容は入出力部 9 に表示出力され、出力された同値クラスについて利用者が入出力部 9 から編集（選択、削除等）を行うことができる。次にテスト項目生成部 4 はこの同値クラスの様式ファイル 3 に列挙された同値クラスの様式でかかれたドキュメントから、同値

クラスを組み合わせることによってテスト項目を生成し、テスト項目の様式ファイル5に出力する。

【0016】一方、内部状態同値クラス生成部6が駆動されると、同値クラス保持部1からの情報やUMLセマンティックス準拠のCASEツール内のクラス図やシケンス図から得られる情報を基にテスト対象メソッド（プログラム）内で必要となるオブジェクトやマスター等と、その同値クラスを同値クラスの様式ファイル3に列举して格納する。この同値クラスの様式ファイル3に格納された内部状態同値クラスのデータは、テスト項目生成部4によって内部状態の同値クラスと上記入力パラメータの同値クラスと組み合わせてテスト項目が生成され、テスト項目の様式ファイル5に格納される。テスト項目の様式ファイル5の内容は、入出力部9から表示出力して利用者（設計者）に提示され、利用者による編集が可能である。

【0017】制御部7はテストの対象としてクラスを指定した時に指定されたクラスのパブリックメソッド（Publicメソッド：他のクラスからアクセスするが可能なメソッド）だけでなく、指定されたクラスのスーパークラス（実装の継承、インターフェースの継承の何れの場合も含む）をUMLセマンティックス準拠のOO-CASEツール8内のクラス図から検出し、そのクラスのパブリックメソッドのテスト項目を生成する。

【0018】

【発明の実施の形態】上記図1に示す本発明の原理構成に示す主要な各部は、CPU、メモリを含むプログラムによる処理部、キーボードやマウス等の入力部、及びディスプレイ等の表示部を備えた情報処理装置によるプログラム処理により実現することができ、以下に各部の機能を実現する処理フローを説明する。

【0019】図3は入力パラメータ同値クラス生成部（図1の2）の処理フローである。この処理では指定されたメソッドの入力パラメータやそれに関連するオブジェクトの同値クラスを、同値クラス保持部とUMLセマンティックス準拠のOO-CASEツールから取得した情報に基づいて作成するものであり、以下に説明する。

【0020】テストを生成する対象となる指定されたメソッド（プログラムのMとする）のシグネチャとしてメソッドの引数とその型、返値の型、メソッドを持つクラスと関連しているクラスとその関連の種類、参照方向、多重度、及びスーパークラスを備える。これらをOO-CASEツールから取得して、メソッドMのシグネチャから引数の数N（総数）を取得する（図3のS1）。次に、引数の番号を表すiを1に設定し（図S2）、iがN以下であるか判別し（図S3）、以下の場合はメソッドMのi番目の引数（この例ではAとする）を取得し（図S5）。続いて、同値クラス情報オブジェクト（この例ではEとする）を作成し（図S6）、Eの仮引数名を上記Aの仮引数名と同じものに設定する（図S7）。

次で、同値クラス作成処理を行う（図3のS8）。この同値クラス作成処理は後に示す図4、図5により内容を説明する。

【0021】この後、上記取得したAの型がオブジェクトであるか判別し（図S9）。オブジェクトで無いと後述するS11へ移行し、オブジェクトの場合はオブジェクトに対する同値クラス取得処理を行う（図S10）。このオブジェクトに対する同値クラス取得処理の詳細は、後に示す図6を用いて説明する。続いて、Aの型が集合（配列）であるか判別し、集合でないとS13へ移行し、集合の場合は集合に対する同値クラスの取得処理を行う（図S12）。この集合に対する同値クラスの取得処理の詳細は後に示す図7を用いて説明する。次いで、iを+1して（図3のS13）、上記S3に戻って同様の処理を行ない、iがNを越えると、入力パラメータ同値クラスの出力処理を行う（図S4）。

【0022】図4は同値クラス作成処理（図3のS8）で実行され、Aの型名、同値クラステーブルを用いる処理のフローである。最初にEの型名（同値クラスの型名）をAの型名（図3のS6で取得）に設定する（図4のS1）。次にスーパークラス取得処理（Aの型名）を行う（図4のS2）。このスーパークラス取得処理（型名A）の詳細は図5に示され、この処理はその基点となつたオブジェクトの同値クラスも利用可能にするための処理であり、そのスーパークラスはここで説明すると、OO-CASEツール（図1の8）からAのスーパークラスSを取得し（図5のS20）、S（スーパークラス）が無いと終了するが、有る場合は同値クラス保持部（図1の1）からクラスSの同値クラスを取得してEの同値クラスに追加し（図5のS22）、スーパークラス取得処理（S）を実行する（図S23）。このスーパークラス取得処理（S）の詳細は図5に示す。

【0023】図4に戻り、スーパークラス取得処理に続いて同値クラス保持部（図1の1）から引数Aの型名をキーにして同値クラスを取得し（図4のS3）、続いて同値クラス保持部からAの型名をキーにして、同値クラスを取得し、それを全てEの同値クラスに追加する（図S4）。続いて、Eを同値クラステーブルに追加し（図4のS5）、同値クラスの作成処理を終了する。

【0024】図8は同値クラステーブルの構造を示す。上記図4の処理により作成された同値クラステーブルは図8に示すように各同値クラス情報が個別に格納され、仮引数名、フィールド名、型名、同値クラスの順にそれぞれの情報が格納されている。

【0025】図6はオブジェクトに対する同値クラス取得処理（図3のS10の処理）のフローである。最初にOO-CASEツールから、クラスAの属性の個数Mを取得し（図6のS1）、属性の個数を表すjを1に設定し（図S2）、jがM以下か判別し（図S3）、M以下でないと終了し、M以下であるとj番目の属性を取得し

(同S4)、同値クラス情報オブジェクトE1を作成する(同S5)。続いて、E1のフィールド名としてAのj番目の属性名を設定し、型名としてAのj番目の属性の型名を設定し(図6のS6)、続いて同値クラス情報オブジェクトE1の同値クラス作成処理(A型名)を行い(同S7、上記図4参照)、jを+1し(同S8)、S3の処理に戻る。

【0026】図7は集合(配列)に対する同値クラス取得処理(図3のS12の処理)のフローである。最初にOO-CASEツールから、クラスAの要素となるものの型Tを取得し(図7のS1)、同値クラス情報オブジェクトE1を作成し(同S2)、E1のフィールド名としてAの要素を示す名前A[i]を設定し、型名をTとし(同S3)、E1を同値クラステーブル(図8参照)に追加する(同S4)。

【0027】図3に示す入力パラメータ同値クラス生成部の処理フローに対し、入力パラメータがメソッドであり、それが他のオブジェクトと集約関係を備える場合の処理フローを同値クラス生成部の応用例1の処理フローとして図9に示す。

【0028】図9において、最初にOO-CASEツールから引数Aのクラスと集約関係にあるクラス数Nを取得し(図9のS1)、iの値を1に設定し(同S2)、iがN以下か判別し(同S3)、N以下でないと終了するが、N以下の場合は、OO-CASEツールから引数Aのクラスと集約関係にあるクラスのi件目の名前Cとその関連名Rを取得する(同S4)。これを用いて同値クラス情報オブジェクトE1を作成し(同S5)、E1のフィールド名をR、E1の型名をCとし(同S6)、同値クラス作成処理(E1、C)を行う(同S7)。この同値クラス作成処理は上記図4に示されている通りである。次いで、E1を同値クラステーブルに追加し(同S8)、この後のS9～S13の処理は上記図3のS9～S13と同じであり、説明を省略する。この図9の処理により、集約関係にある他のオブジェクトも入力パラメータに加えて、より十分なテスト項目を生成することができ、テスト設計者の負担を軽減することができる。

【0029】次に図10は同値クラス生成部の応用例2の処理フローである。この処理フローは図3に示す入力パラメータ同値クラス生成部の処理において、集約関係をたどる際に関連する他のオブジェクトの方向(一方のオブジェクトから他のオブジェクトを参照または見ることができる場合に他のオブジェクトへの方向)がある場合、その方向がナビゲーション(navigation)として定義され、この図10の応用例2の処理はそのnavigationにより示す関連先のオブジェクトのみを入力パラメータに加えるようにしたものである。

【0030】最初に、OO-CASEツールから引数Aのクラスと集約関係にあるクラス数Nを取得し(図10のS1)、iの値を1に設定し(同S2)、iがN以下

か判別し(同S3)、N以下でないと終了するが、N以下の場合は、OO-CASEツールから引数Aのクラスと集約関係にあるクラスの名前C、その関連名R及び関連の向き(Naviで表す)を取得する(同S4)。次にNaviが引数Aの型であるクラスから、その関連クラスCへ参照可能に設定されているか判別し(同S5)、設定されていないとS14に移行してiを+1する処理を行い、設定されている場合はE1のフィールド名をR、E1の型名をCとし(同S6)、同値クラス作成処理(E1、C)を行う(同S7)。この同値クラス作成処理は上記図4に示されている通りである。次いで、E1を同値クラステーブルに追加し(同S8)、続くS9～S13の処理は上記図3のS9～S13と同じであり、説明を省略する。この図10の処理により、不要なテストを生成することができなくなるため、テスト設計者の負担を軽減することができる。

【0031】図11は同値クラスの出力処理のフローである。

【0032】この処理は、上記図1に示す同値クラスの様式ファイル3に対して実行されて、ディスプレイ上に出力表示される。最初にドキュメント名を出力し(図11のS1)、テスト項目を生成する対象メソッド名を出力し(同S2)、更に空行を出力する(同S3)。続いて、表の名称、表の列名を出力し(図11のS4)、テーブルの長さNを取得し(同S5)、同値クラス出力行の設定を行い(同S6)、変数のiを1に設定し(同S7)、iがN以下か判別して(同S8)、以下でないと表の末尾を示す記号を出力して終了し(同S9)、iがN以下の場合は、同値クラステーブルのi番目の要素をEとして設定し(同S10)、このEの仮引数名、フィールド名、型名を出力する(同S11)。次に同値クラスの個数Mを取得し(図11のS12)、変数jを1に設定し(同S13)、jがM以下か判別して(同S14)、以下の場合は同値クラステーブルi番目の同値クラスのj番目の同値クラスCを取得する(同S15)。続いてCの正／異とCの名前を出力し(同S16、S17)、改行をして(同S18)、jを+1し(同S19)、上記S14に戻って同じ処理を繰り返す。S14において、jがM以下で無い場合は、iを+1して(図11のS20)、S8に戻り、以下の処理を繰り返す。

【0033】図12は同値クラスの出力様式の例を示し、図13は同値クラスのクラス図と同値クラステーブルを示す。

【0034】図12の出力様式は上記図11の出力処理のフローにより出力される。これは入力パラメータ同値クラス表の例であり、内部状態の同値クラスを出力する場合は、「内部状態同値クラス表」となる。図12の入力パラメータ同値クラス表は、図13のA.の構造を持つ「udef」というクラスのオブジェクトと、String(文字列)型のオブジェクトを引数に持つメソッド

ドclass A : methodA (udef obj, String str) の同値クラスを示したものである。すなわち、この入力パラメータ同値クラス表の例では、テスト項目を生成する対象メソッド名として、クラス名がA、メソッド名がAで、最初のオブジェクトの型名が「u d e f」、その引数名が「o b j」であり、次のオブジェクトの型名が「S t r i n g」、その引数名が「s t r 1」である。この図12を内部状態の同値クラスの出力に使用する場合、先頭の列名はオブジェクト名になる。

【0035】図13のA. に示すような関係を持つクラス図について、図13のB. に示す同値クラステーブルが生成される。図13のA. はクラス「object」のサブクラスとして、クラス「udef」が定義されており、「udef」のクラスの属性が「int」（整数）でフィールド名が「attr1」であることを表す。更に、「udef」というクラスは、0～Nの値をとることができる「i n t」というクラスと集約関係を備えていることを表す。このようなクラス図を用いて、図13のB. に示すデータ構造を備えた同値クラステーブルが入力パラメータ同値クラス生成部（図1の2）に保持される。

【0036】図12には先頭の列から仮引数名、フィールド名、型名の各列が設けられ、それぞれに図13のB. に示すテーブルに対応する内容が設定され、「正」、「異」である同値クラスがそれぞれに設定されている。

【0037】図14はテスト項目生成部の処理フローである。この処理は、上記図1のテスト項目生成部4において実行される。最初にテーブルリスト（TLで表す）への同値クラスの読み込み処理を行う（図14のS1）。この処理は上記図3～図10により生成された同値クラスについて行われ、その詳細は、後述する図15、図16に示す。読み込み処理に統いて、ドキュメント名の出力を行い（図14のS2）、テーブルリストの先頭の同値クラステーブル（Tで表す）を取得する（同S3）。次いで、テーブルTのキーKを対象メソッド名として出力し（図14のS4）、空行を出力し（同S5）、表の名称を出力して、テーブルT内の同値クラス情報オブジェクトを用い、同値クラス情報オブジェクトの仮引数名とフィールド名、型名を表の列名として出力する（同S6）。

【0038】続いて同値クラスについて組合せ処理をテーブルリストTLを引数にして処理を行う（図14のS7）。この組合せ処理は後述する図17に示す。組合せ処理が行われると、テーブルリスト（TL）内のテーブルの総数（TTotalで表す）を取得し（同S8）、i=2の設定を行い（同S9）、iが総数（TTotal）以下か判別し（同S10）、総数を越えると処理を終了し、総数以下の場合はテーブルリスト（TL）にTL内のi番目の同値クラステーブルを設定する（同S11）。次いで、テーブル（T）のキーKを対象データ名

として出力し（同S12）、空行を出力し（同S13）、表の名称を出力し、TL内の同値クラス情報オブジェクトを用い、同値クラス情報オブジェクトの仮引数名とフィールド、型名を表の列名として出力する（同S14）。次に上記S14で出力された同値クラス情報についてTLを引数として組合せ処理を実行し（同S15）、i=i+1を実行し（同S16）、S10に戻る。なお、組合せ処理（TL）の詳細は上記S7と同様に後述する図17に示す。

【0039】図15、図16は同値クラス読み込み処理のフロー（その1）、（その2）である。この処理により同値クラスの様式から同値クラステーブルを再構築するための処理である。但しこの処理では、集合を表す型を識別子とし、それを別な同値クラステーブルにしており、出力時には集合は別の表を作成する。

【0040】最初に、空のテーブルリストTLを作成し、同値クラステーブルのカウンタjを1に初期化し（図15のS1）、空のクラステーブルCTjを新規に作成する（同S2）。行数のカウンタiに同値クラス記述されている先頭行数を設定し、Tag（タグ）に同値クラスのメソッド名を設定する（図15のS3）。iが表の最終行か判別し（図15のS4）、最終行でなければ；行目の仮行数名列、フィールド名列のいずれも空かを判別する（同S5）。空きの場合はiに1加算して（図15のS6）、S4に戻る。空きでない場合は、同値クラス情報オブジェクトEを作成し（図15のS7）、そのオブジェクトに対して仮引数名、フィールド名、型名をそれぞれ読み込み、同値クラス情報オブジェクトEに設定する（同S8）。

【0041】Eの型が集合を表しているか否かを判定し（図16のS9）、集合を表していない場合、i行目の正／異列の値と同値クラス列の値をペアにしてEの同値クラスに追加し（同S10）、iに1を加算する（同S11）。統いて、i行目の仮行数名列、フィールド名の列、の両方が空かを判定する（図16のS12）。両方とも空ならば、S10に戻り、どちらかが空でなければ新しい同値クラスを作るため、まず現在使用している同値クラス情報オブジェクトEをCTjに追加し（図16のS13）、S4に戻る。上記S9でEの型名が集合を表していると判定した場合は、CTjとタグをペアにして、テーブルリストTLに追加し（図16のS14）、統いて同値クラステーブルのカウンタjに1加算し（同S15）、同値クラステーブルCTjを新規に作成し（同S16）、タグにEフィールド名を設定し（同S17）、上記S6の処理を行う。

【0042】図17は引数にテーブルリストTLを取る組み合せ処理のフローであり、上記図14 S7において実行される。この処理は、同値クラスを組み合わせてテスト項目を作成するもので、図15、図16で作成したテーブルリストを使用する。また、この処理では、組

み合わせフラグ (f_l_a_g) を使用する。これは、異なる同値クラステーブルの同値クラスを組み合わせるときに、最低限の組み合わせだけを生成するために使用するフラグで、最初に0に設定する(図17のS1)。テーブルリスト内の先頭のクラステーブルを取得し、その中に含まれる同値クラス情報の同値クラスを組み合わせる(同S2)。組み合わせ時に、各々の組み合わせ中の異常な同値クラス数を計数する(図17のS3)。全組み合わせに対して処理を行うため、まず、数の総数(To_tal)を取得し(図17のS4)、カウンタkを1に初期化し(同S5)、kがTo_tal以下か判別し(同S6)、以下であればj番目の組み合わせを取得する(同S7)。

【0043】続いてフラグ (f_l_a_g) が0か判別し(同S8)、0であればj番目の組み合わせと、テーブルリストTL内の他の(先頭以外の)クラステーブルの同値クラスとを組み合わせてテスト項目とする(同S9)。この時、TLの先頭以外のクラステーブルから取得した同値クラスがある範囲を表しているときは、その範囲の境界付近の値(通常は下限に近い値)を選んでテスト項目とする。S9の処理で、先頭以外のクラステーブル内の同値クラスを全て使用したか否かを判定し(図17のS10)、全同値クラスを使用していればフラグ (f_l_a_g) に1を設定する(同S11)。

【0044】その後、kに1加算し(図17のS12)、S6の処理に戻る。S10の処理で全同値クラスを使用していないときは、フラグ (f_l_a_g) はそのまままでS12の処理を行う。S8でフラグ (f_l_a_g) が1だった場合、j番目の組み合わせと、テーブルリストTL内の他の(先頭以外の)クラステーブルの正常な同値クラスとを組み合わせてテスト項目とする(図17のS13)。この時、テーブルリストTLの先頭以外のクラステーブルから取得した同値クラスがある範囲を表しているときは、その範囲の境界付近の値(通常は下限に近い値)を選んでテスト項目とし、その後、S6の処理に戻る。上記S6において、kがTo_talよりも大きくなったら組み合わせ処理を終了して出力のための処理S14を実行する。

【0045】図18は生成されたテスト項目の具体例であり、上記図14のテスト項目生成部の処理により生成される。

【0046】図18のA. はテスト項目表で、テスト項目を生成する対象メソッド名が、classA:methodA(objec t obj, string str1) である例である。この例は、入力パラメータのテスト項目であり、仮引数名、型名、同値クラスの組み合わせが列出されている。具体的には、「obj」、「obj.att1」、「str1」及び「coll」という同値クラスの各値を一つずつ取り出して組み合わせて、A. に示すような各種の組み合わせが得られる。長さについては、文字列の長さを下限(長さ0、長さ2等)にす

るか、最大長としている。図18のB. は対象データが集合(配列)である「coll[i]」という名称の入力パラメータのテスト項目を表し、整数の配列が集まっているもので長さが2以上の場合にはB. に示した表を参照してテスト項目を作成する。

【0047】図19、図20は内部状態の同値クラス生成部(図1の6)の処理フロー(その1)、(その2)である。この処理フローは指定されたプログラム(メソッド)内で参照・更新するオブジェクトやそれに関連するオブジェクトの同値クラスを、同値クラス保持部(図1の1)とUMLセマンティックス準拠のOO-CASEツールから取得した情報に基づいて作成するものである。すなわち、プログラムで参照されるデータが入っているデータベース(内部状態を表すデータを保持)としてどのようなデータを用意すべきか、必要な状態の組み合わせを作るための処理である。

【0048】最初にOO-CASEツールから、テストを生成するメソッドMを含むUMLに従ったシーケンス図(処理の様子を表す図)を検索して、メソッドの動作に関連しているオブジェクトを検出して取得する(図19のS1)。次にメソッドMを含むシーケンス図の数Nを取得し(同S2)、i=1の設定を行い(同S3)、iがN以下か判別し(同S4)、N以下の場合はi番目のシーケンス図Sを取得し(同S5)、シーケンス図S中に現れるオブジェクトの数Lを取得し(同S6)、j(jは図示省略された処理でj=1に初期化されているものとする)がL以下か判別し(同S7)、L以下の場合はシーケンス図S中にj番目に現れるオブジェクトoを取得する(同S8)。次にオブジェクトoは、メソッドMのシグネチャ(メソッドの名称、引数の型、返値の型等)に含まれるか判別し(図19のS9)、含まれない場合はそのオブジェクトoを候補リストに加えて(同S10)からS11に移行し、含まれる場合はS11に移行して、jを+1して(同S11)、S7に移行する。S7においてjがL以下でない場合はiを+1して(図19のS12)、上記S4に移行する。

【0049】S4において、iがN(シーケンス図の数)を越えると、候補リストの中のオブジェクトの数Kを取得し(図20のS13)、空の同テーブルCTを作成し(同S14)、iを1に設定し(同S15)、iがK以内か判別し(同S16)、以下であれば候補リスト中のi番目のオブジェクトoを取得する(同S18)。次に同値クラス保持部(図1の1)からオブジェクトoの型名をキーにして同値クラスを取得し(図20のS19)、同値クラス情報オブジェクトEを作成する(同S20)。次に同値クラス情報オブジェクトEの仮引数名をoのクラス名に設定し(図20のS21)、オブジェクトEの同値クラス作成処理(o. クラス名)を行う(同S22)。この同値クラス作成処理は、上記図4、図5に示したものと同様である。

【0050】続いて、○の型がオブジェクトか判別し（図20のS23），オブジェクトであればオブジェクトに対する同値クラス取得処理（上記図6参照）を行い（同S24），オブジェクトでない場合又は前記S24の処理が行われた後、同値クラス情報オブジェクトEを同値クラステーブルCTに追加し（同S25），iを+1して（同S26），S17に移行し内部状態同値クラスの出力処理を行って、上記S16に戻る。また、S16において、iがk（候補リストの中のオブジェクトの数）を越えた場合も上記S17の処理を行う。

【0051】この図19、図20の処理により、テスト対象メソッドが出現するシーケンス図の中に、テスト対象のメソッドの入力パラメータや返値のオブジェクト以外のものをメソッドの動作に関連するオブジェクトの候補とする。これらの候補の中には、他オブジェクトに必要なものも含まれ、この処理で生成された同値クラス表は、入力パラメータと同様に修正可能であり、また、テスト項目生成部によって内部状態のテスト項目を作成することができる。内部状態のテスト項目を入力パラメータのテスト項目と組合わせることによって、より詳細なテスト設計が行える。また、内部状態のテスト項目は、テスト用に作成するデータベースにどのようなレコードを用意すればよいかの指針になる。

【0052】上記した各処理では、メソッドをテスト対象として指定して、そのメソッドについての入力パラメータやデータベースの状態を作るものであったが、クラスを指定してその中のメソッドの全ての操作についてのテストを行うようにしたい。その制御が制御部により実現することができ、以下に説明する。

【0053】図21はクラス指定による制御部（図1の7）の処理フローである。この処理フローは、指定されたクラスと、そのスーパークラスのpublicメソッド（外部から呼び出すことができるメソッドであり、内部操作で使用するプライベートメソッドと区別される）をOO-CASEツールから取り出し、各々のpublicメソッドに対して、同値クラスの生成、テスト項目の生成を行う処理を示している。

【0054】すなわち、最初にテスト項目を生成するクラスCを入力し（図21のS1），OO-CASEツール（図1の8）から、クラスCに含まれるpublicメソッドの個数Nを取得し（同S2），メソッド名リスト（MLとする）を設け（同S3），iを1に設定し（同S4），iがN以下か判別し（同S5），iがNを越えた場合はS11に移行して、クラスCのスーパークラスに対して本処理（図21に示す「クラス指定による制御部の処理」）を行う。S5において、iがN以下である場合は、OO-CASEツールから、クラスCに含まれるi番目のpublicメソッドMを取得する（同S6）。次いで、メソッドMに対して、入力パラメータの同値クラス生成部の処理を呼び出し、同値クラスの様式M1を取得

し（図21のS7），メソッドMに対して、内部状態の同値クラス生成部の処理を呼び出し、同値クラスの様式M2を出力し（同S8），メソッド名Mをメソッド名リストMLに追加し（同S9），iを+1して（同S10），上記S5に移行する。

【0055】このように、クラス指定により指定されたクラスの全パブリックメソッドに対してテスト項目を作成することができ、オブジェクト指向の性質を考慮したテスト設計の支援をすることが可能となる。

【0056】

【発明の効果】本発明によれば、設計ドキュメントをUMLセマンティックス準拠のOO-CASEツールで作成していれば、その中に含まれる情報を用いてテスト設計を支援することができる。本発明による同値クラス保持部の情報を作成することは、テスト項目を直接作成することより容易であり、ここで十分な同値クラスを抽出しておけば、生成されるテスト項目がより良いものになる。

【0057】また、生成されるテスト項目の数が多い場合は、テスト項目の中から実行可能な数だけ抽出して実行することになる。これは、グループ内で選択の方針を決めておけば、アプリケーションに精通していない人でも必要、十分なテスト項目を選択することが可能となる。

【図面の簡単な説明】

【図1】本発明の原理構成を示す図である。

【図2】同値クラス保持部で保持するデータの構成例を示す図である。

【図3】入力パラメータ同値クラス生成部の処理フローを示す図である。

【図4】同値クラス作成処理のフローを示す図である。

【図5】スーパークラス取得処理（型名A）の詳細を示す図である。

【図6】オブジェクトに対する同値クラス取得処理のフローを示す図である。

【図7】集合（配列）に対する同値クラス取得処理のフローを示す図である。

【図8】同値クラステーブルの構造を示す図である。

【図9】同値クラス生成部の応用例1の処理フローを示す図である。

【図10】同値クラス生成部の応用例2の処理フローを示す図である。

【図11】同値クラスの出力処理のフローを示す図である。

【図12】同値クラスの出力様式の例を示す図である。

【図13】同値クラスのクラス図と同値クラステーブルを示す図である。

【図14】テスト項目生成部の処理フローを示す図である。

【図15】同値クラス読み込み処理のフロー（その1）

を示す図である。

【図16】同値クラス読み込み処理のフロー（その2）
を示す図である。

【図17】組合わせ処理のフローを示す図である。

【図18】生成されたテスト項目の具体例を示す図である。

【図19】内部状態の同値クラス生成部の処理フロー（その1）を示す図である。

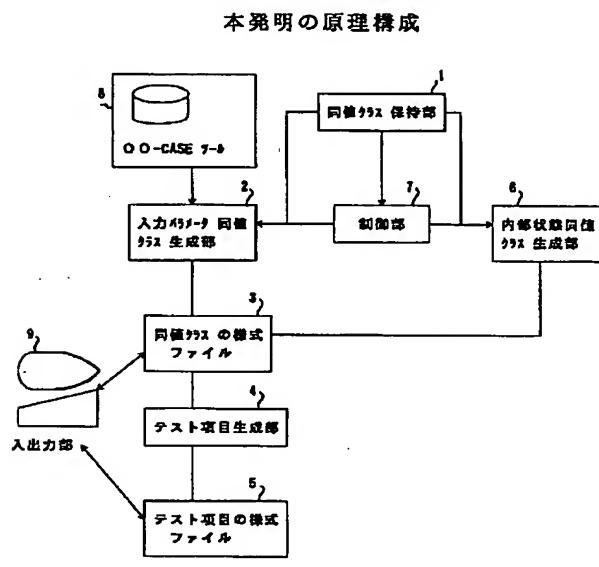
【図20】内部状態の同値クラス生成部の処理フロー（その2）を示す図である。

【図21】クラス指定による制御部の処理フローを示す図である。

【符号の説明】

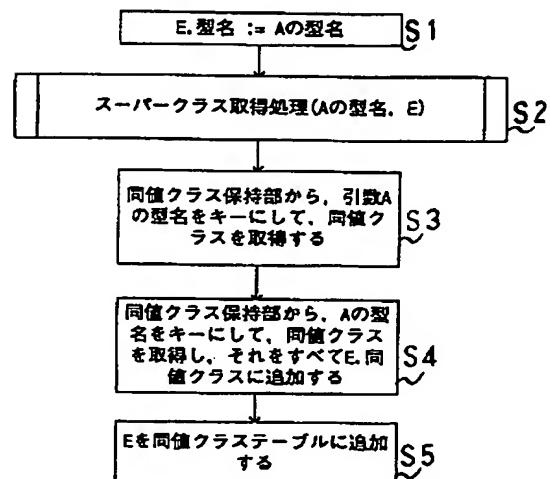
- 1 同値クラス保持部
- 2 入力パラメータ同値クラス生成部
- 3 同値クラスの様式ファイル
- 4 テスト項目生成部
- 5 テスト項目の様式ファイル
- 6 内部状態同値クラス生成部
- 7 制御部
- 8 UMLセマンティックス準拠のOO-CASEツール
- 9 入出力部

【図1】



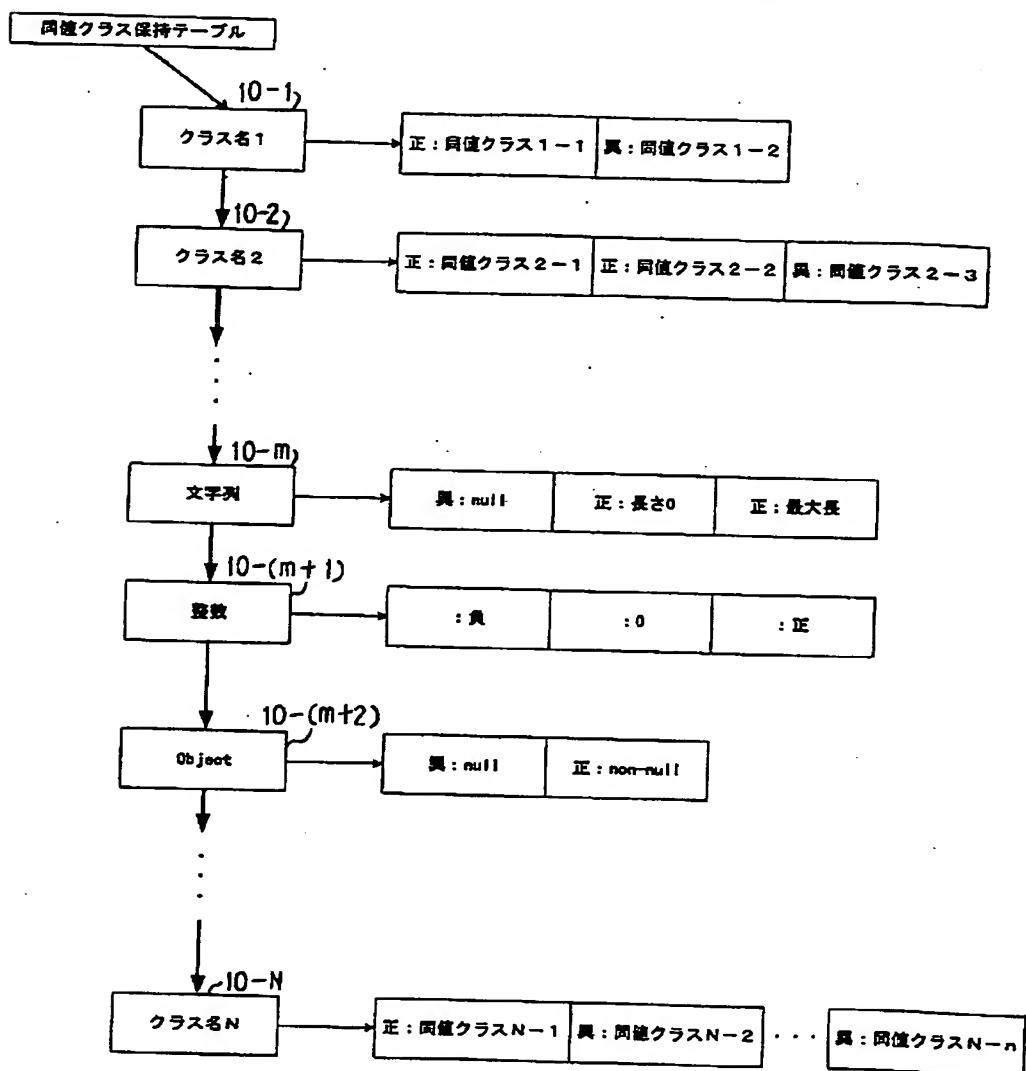
【図4】

同値クラス生成処理のフロー



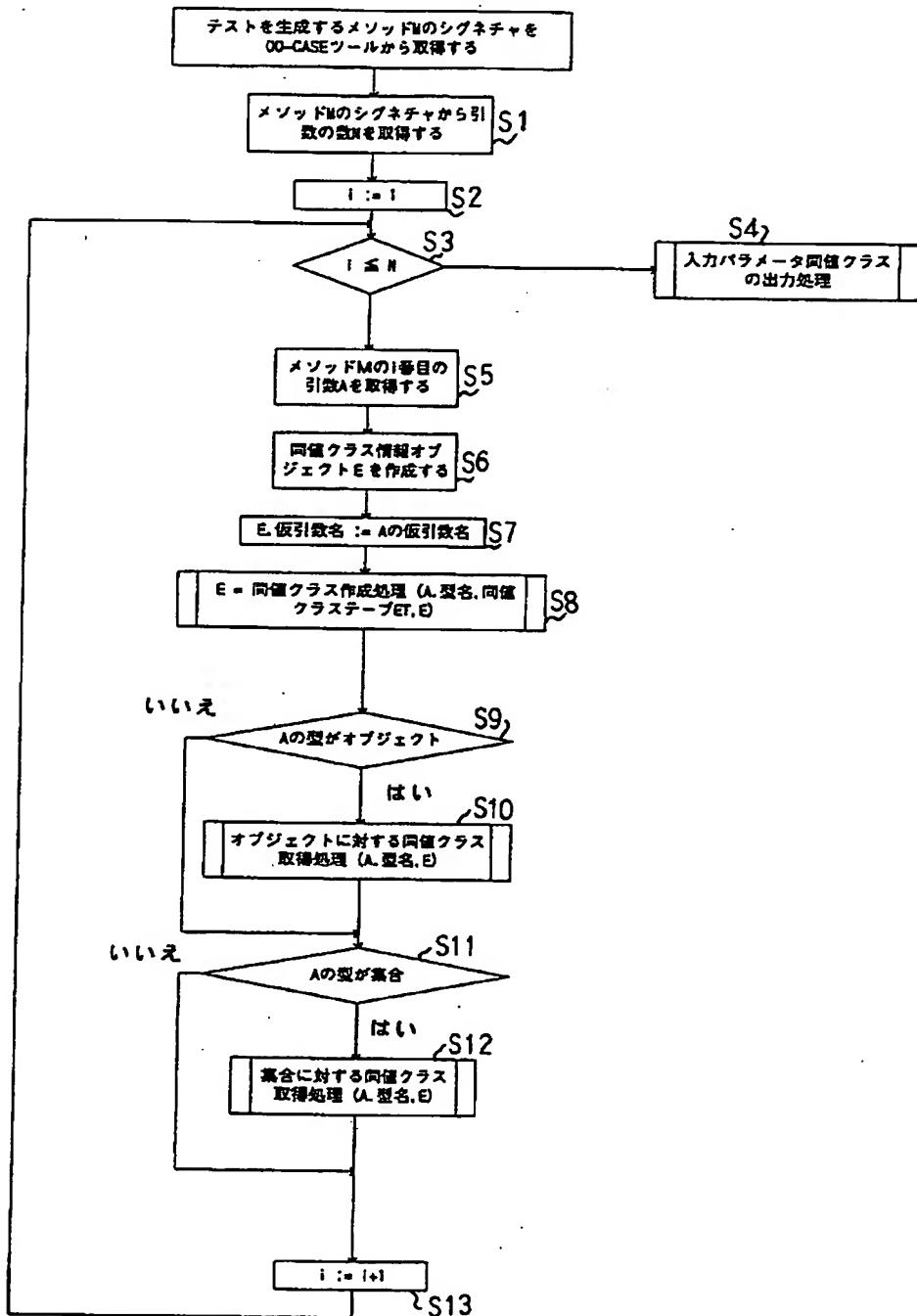
【図2】

同値クラス保持部で保持するデータの構成例



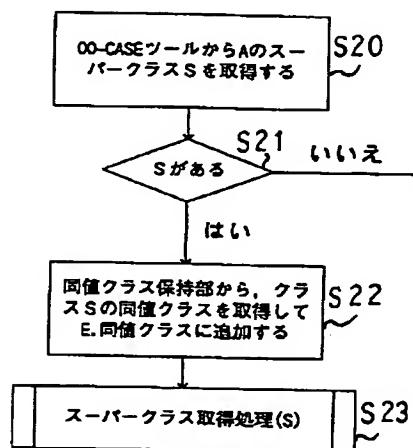
【図3】

入力パラメータ同値クラス生成部の処理フロー



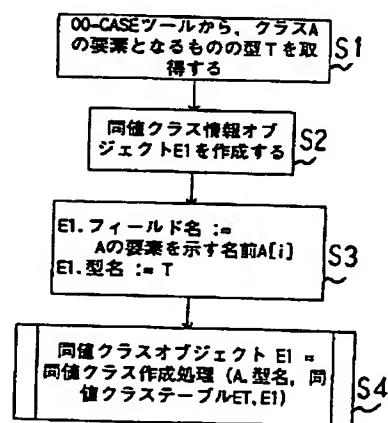
【図5】

スーパークラス取得処理(型名A)の詳細



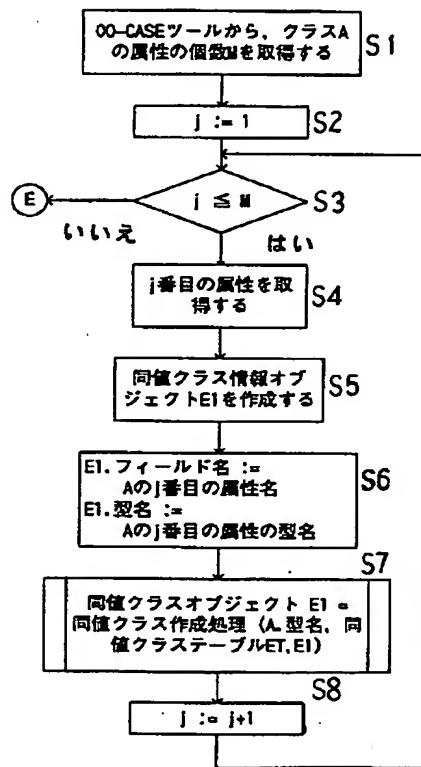
【図7】

集合(配列)に対する同値クラス取得処理のフロー



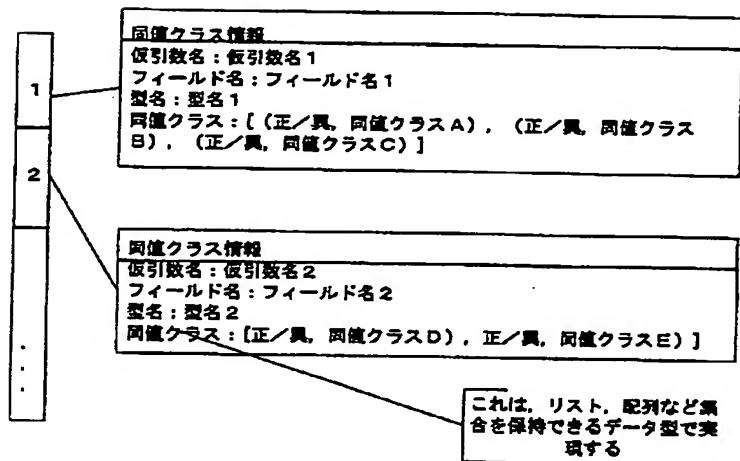
【図6】

オブジェクトに対する同値クラス取得処理のフロー



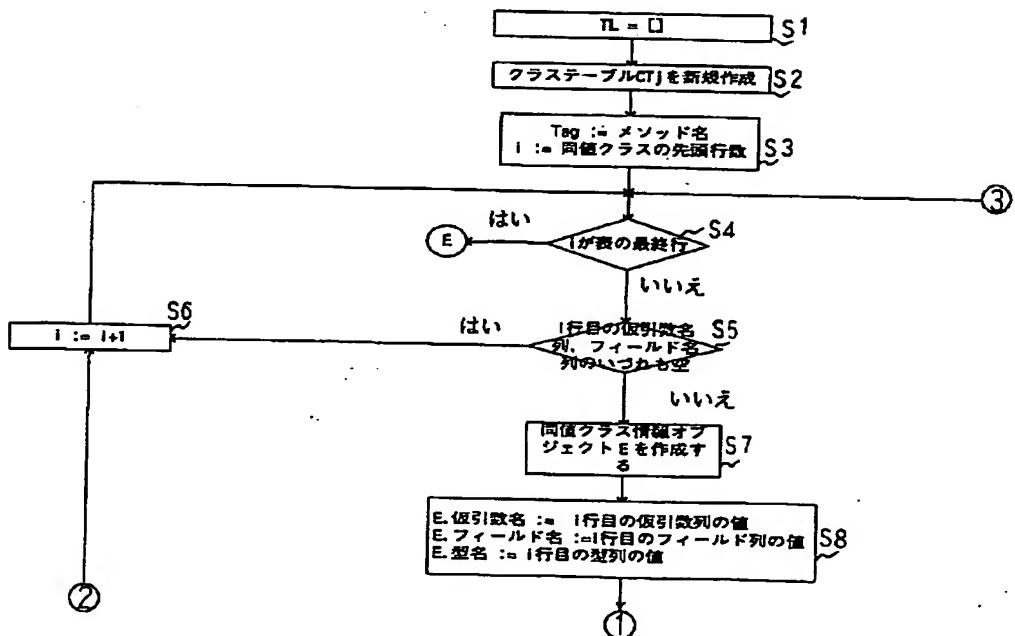
【図8】

同値クラステーブルの構造



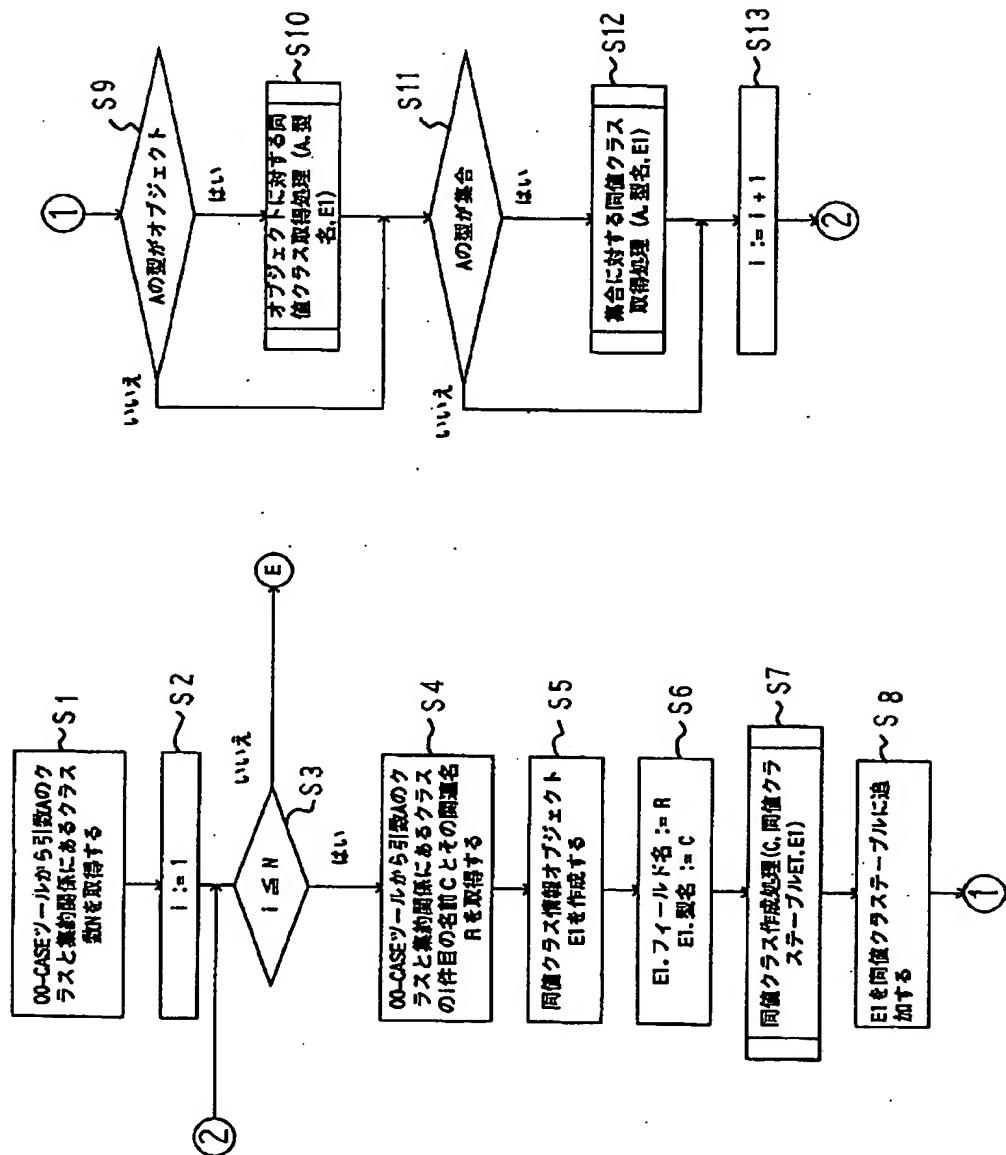
【図15】

同値クラス読み込み処理のフロー (その1)



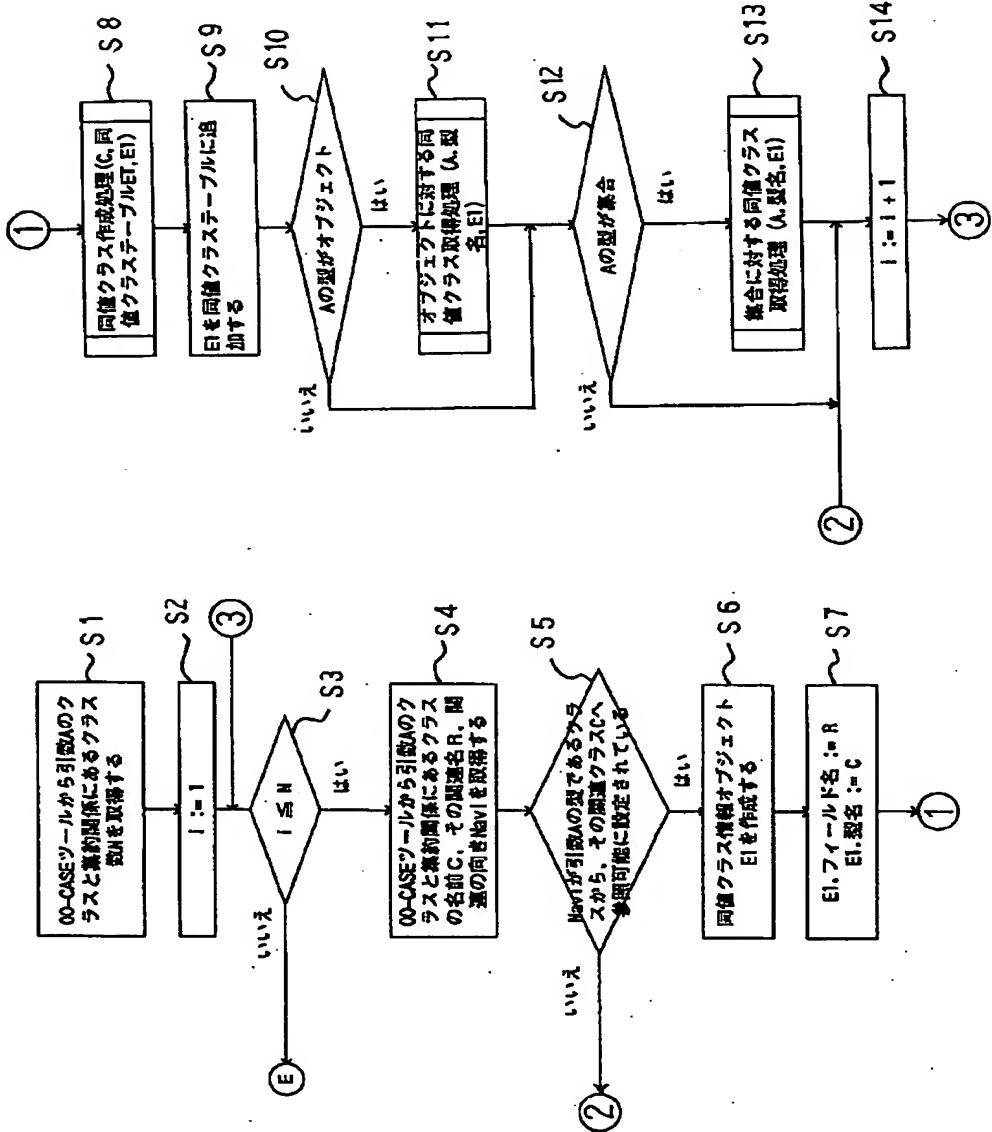
【図9】

同値クラス生成部の応用例1の処理フロー



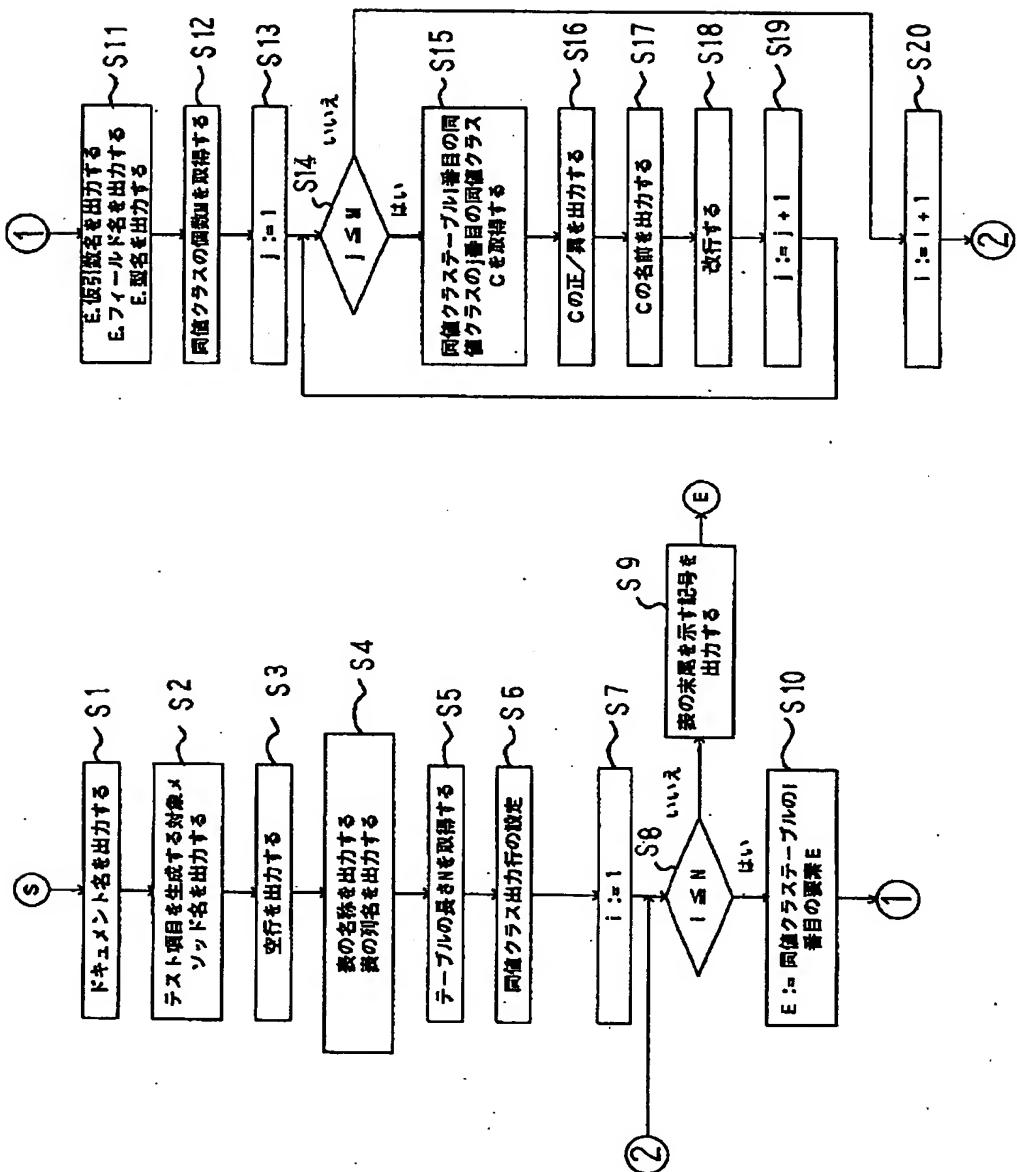
【図10】

同値クラス生成部の応用例2の処理フロー



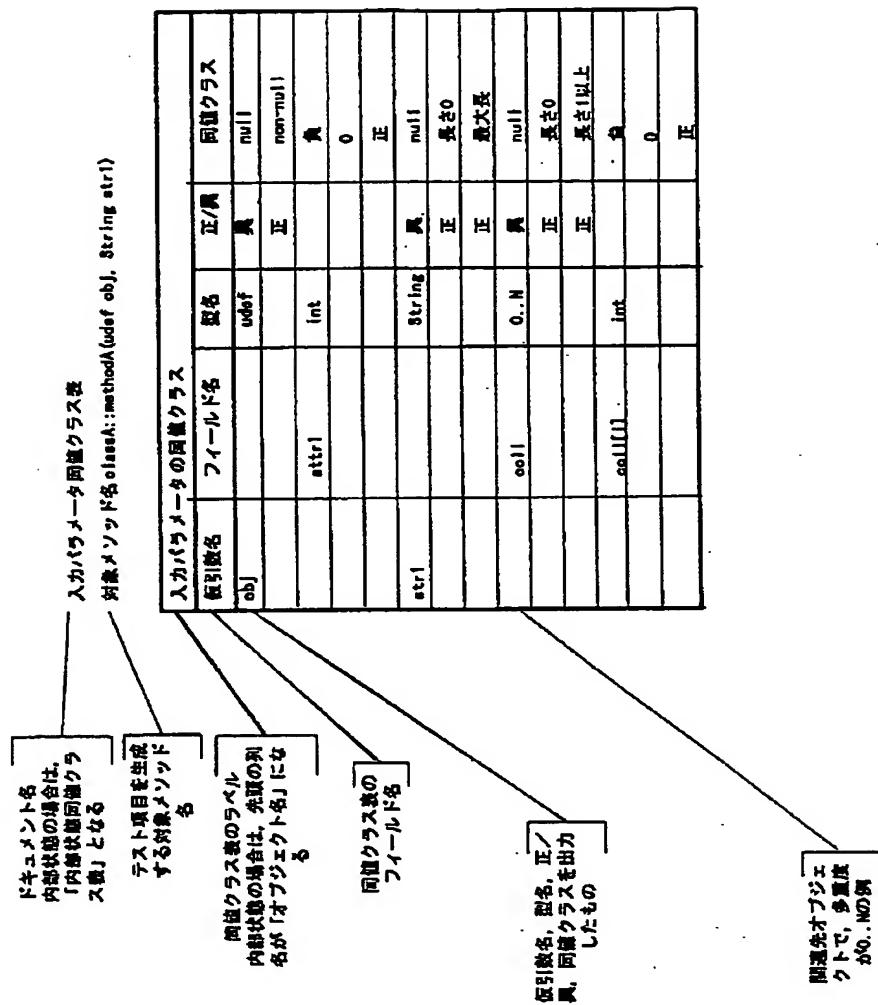
【図11】

同値クラス出力処理のフロー



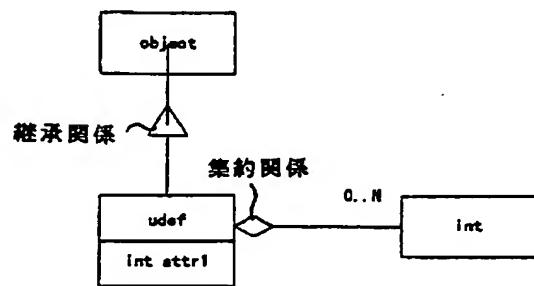
【図12】

同値クラスの出力様式の例



【図13】

同値クラスのクラス図と同値クラステーブル



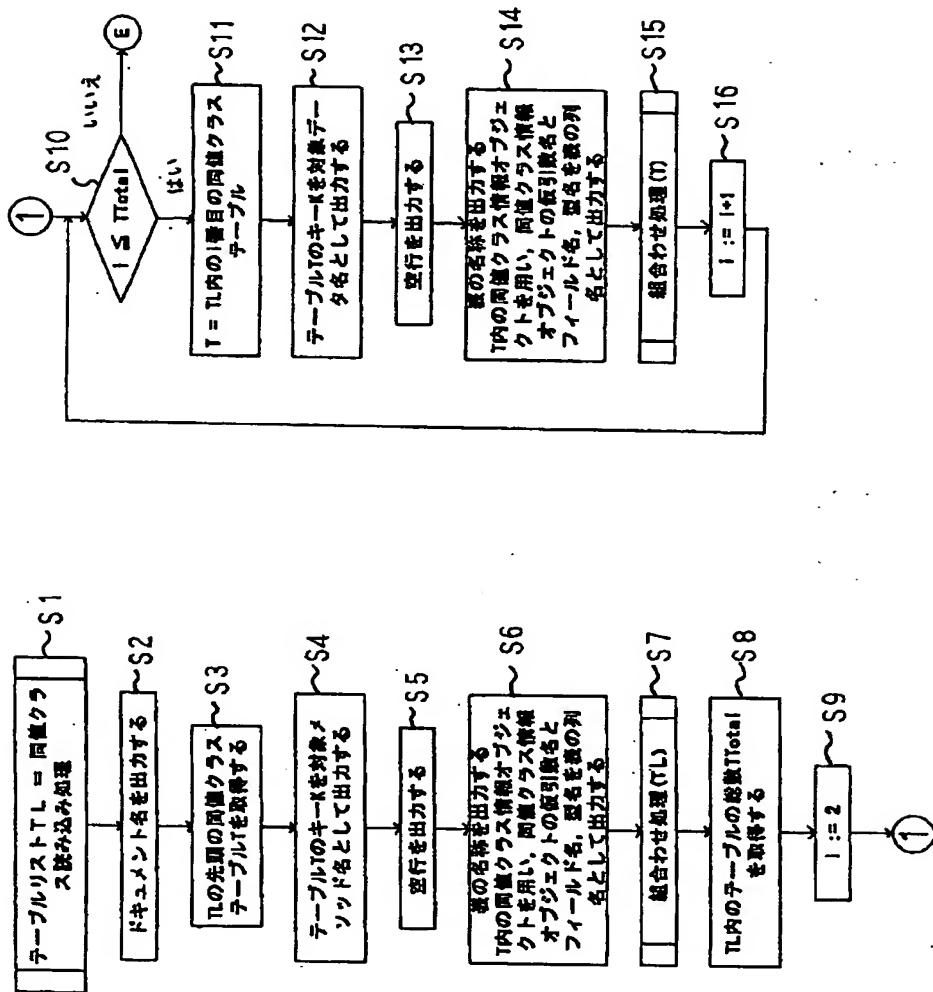
A.

仮引数名 : ob フィールド名 : null 型名 : udef 同値クラス : [(異. null), (正. non-null)]
仮引数名 : null フィールド名 : attr1 型名 : int 同値クラス : [(し負), (し0), (し正)]
仮引数名 : str1 フィールド名 : null 型名 : String 同値クラス : [(異. null), (正. 長さ0), (正. 長さ1以上)]
仮引数名 : null フィールド名 : coll1 型名 : O..N 同値クラス : [(異. null), (正. 長さ0), (正. 長さ1以上)]
仮引数名 : null フィールド名 : coll1[i] 型名 : int 同値クラス : [(し負), (し0), (し正)]

B.

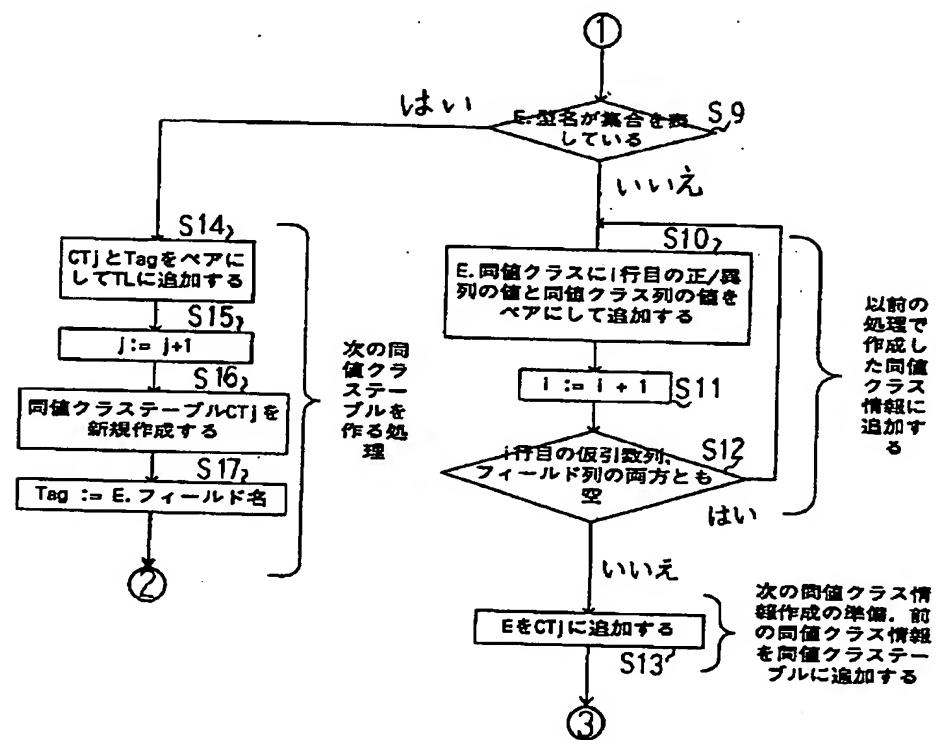
【図14】

テスト項目生成部の処理フロー



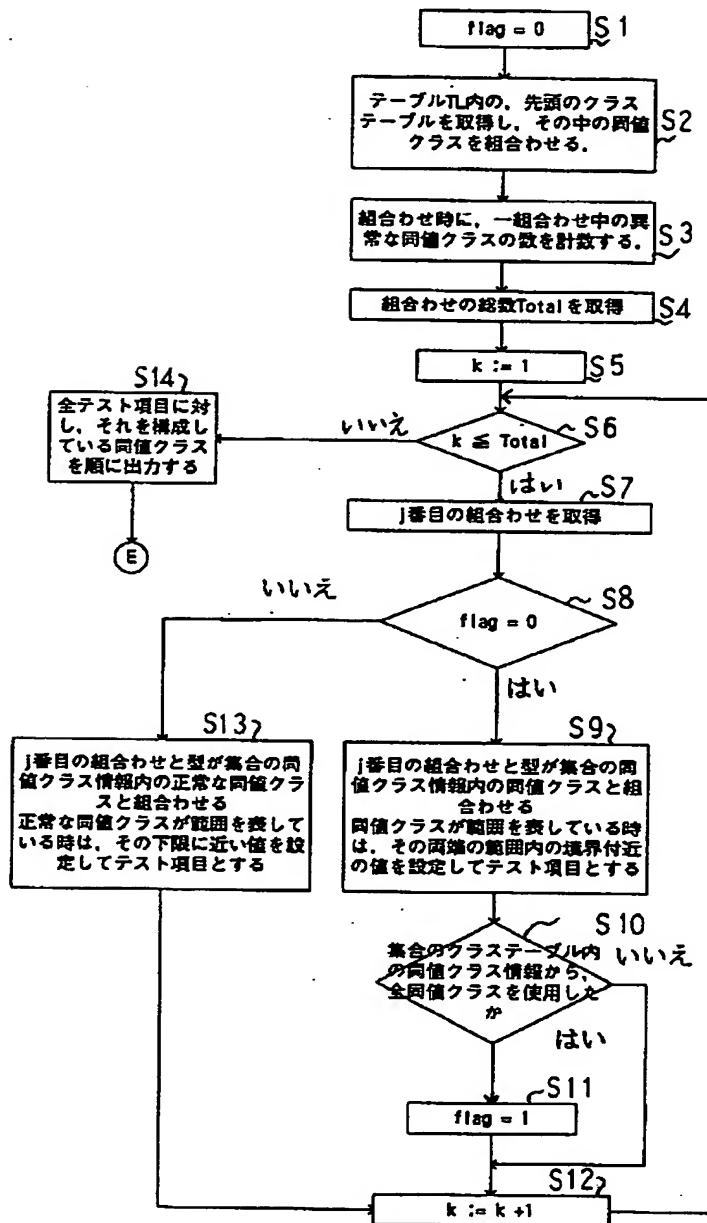
【図16】

同値クラス読み込み処理のフロー (その2)



【図17】

組合せ処理のフロー



【図18】

生成されたテスト項目の具体例

テスト項目表
対象メソッド名 classA::methodA(object obj, String str1)

表のラベル
仮引数名
仮引数名, 型名, 同様クラスを出力したもの

入力パラメータのテスト項目					
仮引数名	obj	obj.attr1	str1	coll	異常数
型	object	int	String		
non-null	負	長さ0	null	0	
non-null	負	長さ0	長さ0	0	
non-null	負	長さ0	最大長	0	
non-null	負	最大長	長さ2	0	
non-null	0	長さ0	長さ2	0	
non-null	0	最大長	長さ2	0	
non-null	正	最大長	長さ2	0	
non-null	正	長さ0	長さ2	0	
non-null	負	null	長さ2	1	
non-null	0	null	長さ2	1	
non-null	正	null	長さ2	1	
null	0	長さ0	長さ2	1	
null	0	最大長	長さ2	1	
null	負	長さ0	長さ2	1	
null	負	最大長	長さ2	1	
null	正	長さ0	長さ2	1	
null	正	最大長	長さ2	1	
null	0	null	長さ2	2	
null	負	null	長さ2	2	
null	正	null	長さ2	2	

A.

対象データ名 coll[i]

入力パラメータのテスト項目		
仮引数名	coll[i]	異常数
型	int	
	負	0
	0	0
	正	0

B.

【図19】

内部状態の同値クラス生成部の 処理フロー（その1）

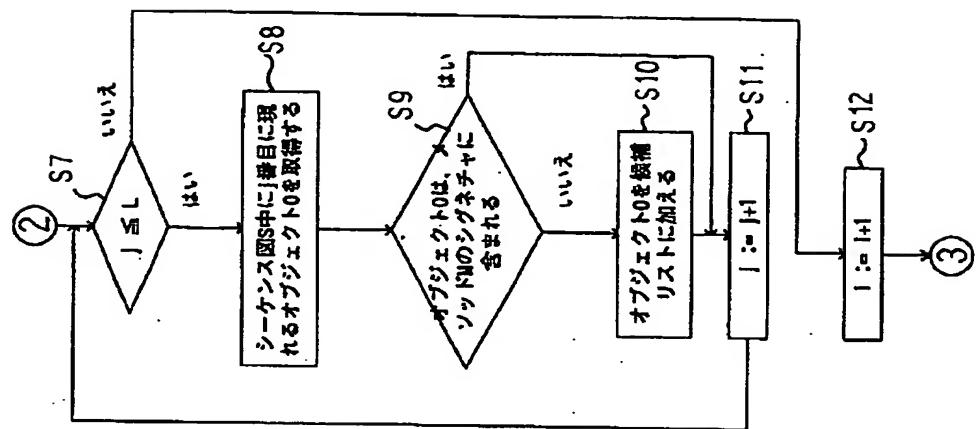
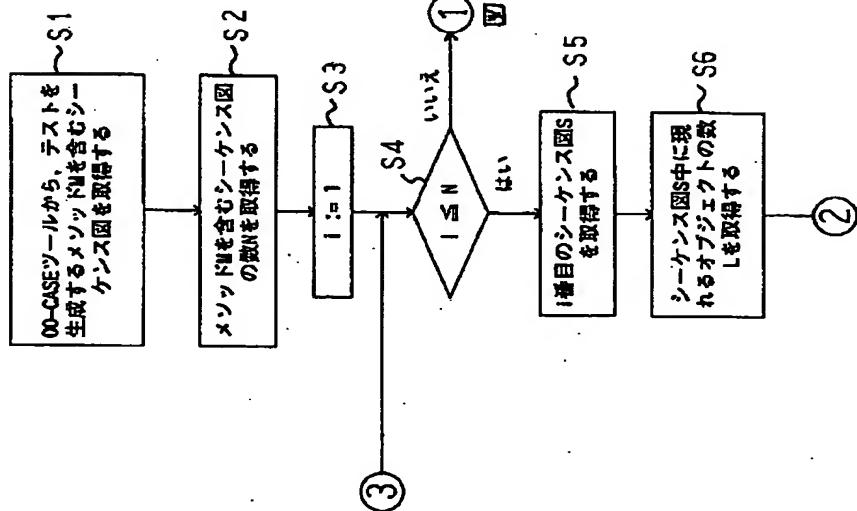
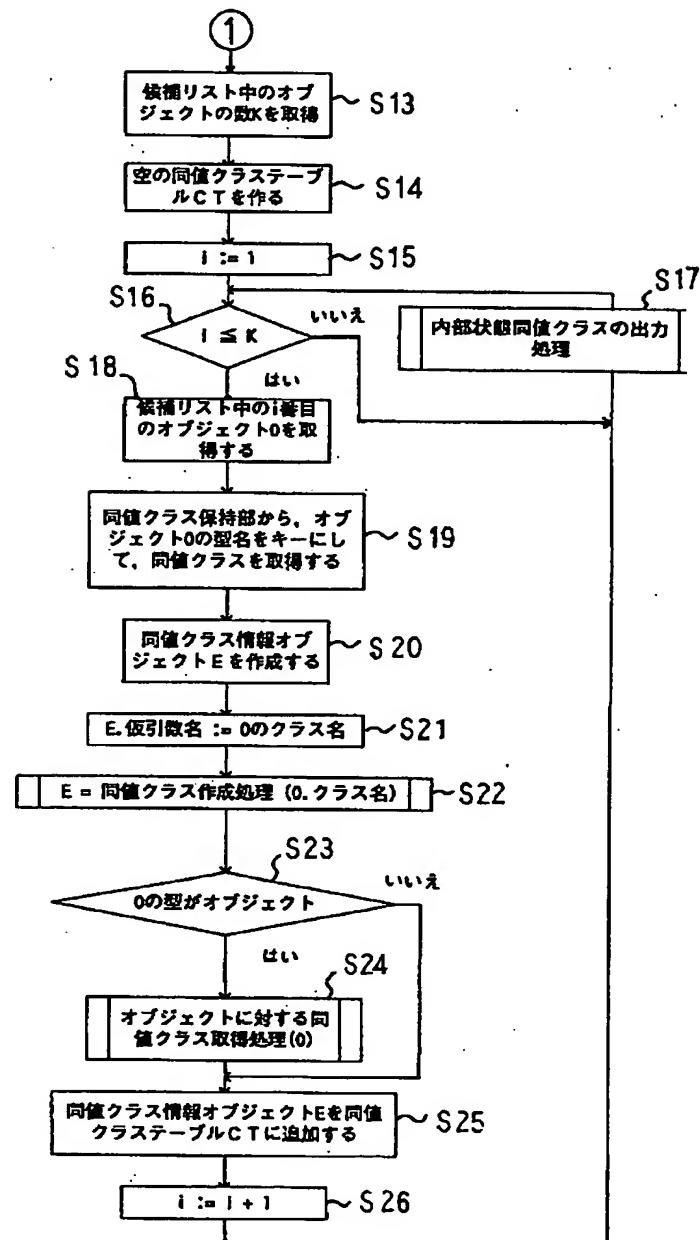


図20へ



【図20】

内部状態の同値クラス生成部の 処理フロー（その2）



【図21】

クラス指定による制御部の処理フロー

